

浪脚本零起点入门系列(八)

maxscript 全局变量、局部变量及注释

作者：飞浪

声明:本教程为 **CG++** 原创,转载请注明出处,谢谢合作:)

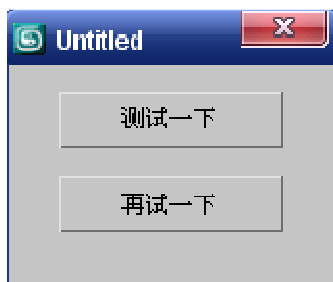
本节关键词:

全局变量 `global` 局部变量 `local` `messagebox` `destroydialog`

在第七节里, Dorff999 提出了这样一个问题: 当一个移转的变数资料,若超过了 `rollout(..)` 包含的内容外,该怎样才能继续存在这个值呢?他想到了用 `global`, 对的,就是这个。这一节我们就来看看全局变量与局部变量的区别,还有一些变量命名的约定俗成的规则,有些是我个人总结的经验,如果有不同看法欢迎提出。

全局变量,通俗点,就是你的整个脚本都可以用的一个变量。**局部变量**,就是指在一定范围才能使用的变量。怎么定义呢?我们通过下面这个例子来深刻了解一下:

```
rollout unnamedRollout "Untitled" width:162 height:108
(
button btn3 "测试一下" pos:[25,13]
    width:111 height:28
button btn5 "再试一下" pos:[25,56] width:111 height:27
on btn3 pressed do
(
)
on btn5 pressed do
(
)
)
createdialog unnamedRollout
运行后得到这样一个界面,
```



这里我们只做了两个按钮来做下测试。

现在加两个变量，如下：

```
global str1 = "这是一个全局变量"
rollout unnamedRollout "Untitled" width:162 height:108
(
local str2 = "这是一个局部变量"
button btn3 "测试一下" pos:[25,13] width:111 height:28
button btn5 "再试一下" pos:[25,56]
    width:111 height:27
on btn3 pressed do
(
    messagebox str1
)
on
    btn5 pressed do
(
    messagebox str2
)
)
createdialog unnamedRollout
```

上面命名有两个字符变量，**str1** 和 **str2**，凡是变量命名用 **global** 声明了的都是全局变量。整个脚本的任何地方都能对其进行修改调用。凡是变量名用 **local** 声明了的，就是局部变量，它只能在当前所在的范围(可以理解为当前语句的括号)内使用。

上面有一句：**messagebox str1**，这句是生成一个弹出信息窗口，简单用法是

messagebox <字符串> 详细参考见：

http://www.cgplusplus.com/online-reference/maxscript-reference/source/maxscript_message_and_query_dialogs.htm

上面脚本运行后，单击生成的按钮，就会看到结果。下面来改一下代码：

```
global str1 = "这是一个全局变量"
rollout unnamedRollout "Untitled" width:162 height:108
(
local str2 = "这是一个局部变量"
global str4 = "这个呢? "
button btn3 "测试一下" pos:[25,13] width:111 height:28
button btn5 "再试一下" pos:[25,56]
    width:111 height:27
on btn3 pressed do
(
    messagebox str1
)
    on btn5 pressed do
(
    createdialog test2
)
)
rollout test2 "Untitled2" width:162 height:57
(
local str5 = "第五句测试! "
button btn8 "再来试下" pos:[22,13] width:125 height:27
on btn8 pressed do
(
    messagebox str4
)
)
createdialog unnamedRollout
```

另外加了一个 rollout，这样可以通过第一个 rollout 打开第二个，就是在按钮里面

写上 `createdialog test2`

我们运行试一下，单击第二个按钮，出现什么？第一次运行会有错误。为什么呢？因为程序从上到下开始运行，当你要点那个按钮 `btn5` 时，里面有个变量 `test2` 在前面没有定义过，它是什么呢？不知道，所以程序运行出错。这时候如果你运行第二次，结果又是对的，又为什么呢？因为你运行一次后，`test2` 就当成全局变量定义了，还在内存里面，下次可以继续用。要避免这种情况也很简单，改一下第一句：

```
global str1 = "这是一个全局变量",test2
rollout .....
```

这样，无论怎么运行不会出错，因为 `test2` 先定义了是个全局变量。在有申明 `global` 或 `local` 的时候，可以用半角逗号来分隔每一个定义的变量。再回来看，刚才那个程序里面，`rollout test2` 里面调用了 `rollout unnamedRollout` 里面一个变量 `str4`，这样也没出错，因为 `str4` 前面用 `global` 申明了，**全局变量 `global` 可以在任意位置定义**。但是如果你把 `test2` 里面的 `str4` 改成 `str2` 就不行了，因为 `str2` 只能用在 `unnamedRollout` 里面。除非，你在 `test2` 里面再定义一次 `str2`。

全局变量很好理解，这个变量名整个脚本通用，所以具有唯一性，就像全球通：D，而局部变量就像本地电话卡，出了地区就不能用了，除非你再去弄一个。局部变量向里通用，我们写的程序都是一对对的括号括起来的，在一个括号里面申明的局部变量，在这个括号里面任何位置可以用，出了这个括号就不能用。我们讲变量可以用 `global`,`local` 申明，可能就有人嫌麻烦了。其实，**脚本能自动识别一个变量是全局的还是局部的，如果我们不用 `global` 或 `local` 也可以**。脚本就会根据变量出现的位置自动判断此变量属于全局的还是局部的。如：

```
str1 = "这是一个全局变量"; test2
.....
```

```
createdialog unnamedRollout
```

这个时候，每个变量就不能用逗号了，没有申明，变量赋值就是单独的语句，要么换行写，要么用 `;` 号表示语句结尾。去掉所有申明，那么刚才那个 `str4` 就变成局部变量了，因为它躲在里面。而 `rollout` 的名称 `unnamedRollout` 和 `test2` 都默认为全局变量。`rollout` 里面开头那个变量去掉 `local` 会出错。我们用按钮测试：

```
on btn3 pressed do
(
str6 = "这变量只能在这个按钮里面才能用！"
```

messagebox str1

)

全局变量可以通用，这个概念是，只要你运行过脚本的 3dmax 没有关闭，那么全局变量还在内存里面，如果别人的脚本跟你用的全局变量名相同，就有可能出现错误。比如如果都用 myrollout 定义一个 rollout，说不定用别人脚本打开的 rollout 是你定义的，为避免这种情况，所以建议变量命名时，形成自己独有的习惯，尽量不要跟别人重复

，而且变量名最好一看就大概明白其含义。如果看过本人写的脚本的朋友可能会发现，如 **VRay 代理管理器**：

<http://www.cgplusplus.com/bbs/viewthread.php?tid=203>

我的命名 global fl_vrayproxyMgr, fl_collectVRayProxies，几乎我的变量都在前面加一个 fl_，就是飞浪缩写，然后后面再加上变量的具体意义，如 vrayproxymgr = vrayproxy manager 就是 vray 代理管理器，collectVRayProxies 就是收集 vray 代理，这样我不看程序也知道我的变量是干什么的，对于代码修改维护具有很大的帮助，而且对于我这样的变量名，你想重复估计可能性不大。特别对于 rollout 这样的默认全局变量名，最好不要简单了事。其实还有一个办法避免全局变量名重名，就是把你的所有程序代码括起来，让他们都变成局部变量，嘿嘿。

注释，每个程序都有的一种重要的功能，用来解释代码，**没有注释的程序代码是不完整的**。也许你不习惯加注释，不过建议养成这个习惯，不然过些时间连你自己都忘记自己的代码是什么意思了。怎么加注释呢？maxscript 提供了两种办法：

一种是用英文的两个 "-" 号，如：--这是一个注释，当然两个以上的也是注释了。

另一种是与 C++ 相同的方法：/*这个里面是注释*/

这两个各有各的好处，"--"可以随意增加，更方便小型注释，/*.....*/适用于大篇注释。把刚才那个程序加注释最后，每次运行上面那个代码就弹出一个对话框，可以加一句关掉相同的以前的对话框，如下：

try destroydialog unnamedRollout catch()

try ... catch 语句意思是，如果运行出错，则不报错，继续执行代码，destroydialog 是与 createdialog 对应的，就是关闭对话框，大家有时间自己查下参考文档，在线参考地址为：

try ... catch :

[url=]http://www.cgplusplus.com/online-reference/maxscript-reference/source/try_expression.htm[/url]

destroydialog :

<http://www.cgplusplus.com/online-reference/maxscript-reference/source/destroydialog.htm>

加了注释的完整示例如下

```
/*
本脚本是飞浪脚本零起点入门系列(八)的一个示例，本节讲的是：
maxscript 全局变量、局部变量及注释
*/

try destroydialog unnamedRollout catch()

-----全局变量-----

global str1 = "这是一个全局变量", test2

-----界面-----

rollout unnamedRollout "Untitled" width:162 height:108
(
    local str2 = "这是一个局部变量"--注释如果放到引号里面就不行了
    global str4 = "这个呢？"

    button btn3 "测试一下" pos:[25,13] width:111 height:28
    button btn5 "再试一下" pos:[25,56] width:111 height:27

    on btn3 pressed do
    (
        --测试一下局部变量

        str6 = "这变量只能在这个按钮里面才能用！"

        messagebox str1
    )

    on btn5 pressed do
    (
        createdialog/*这种注释可以随便加*/ test2
    )
)

rollout test2 "Untitled2" width:162 height:57
```

```
(  
    local str5 = "第五句测试！"  
  
    button btn8 "再来试下" pos:[22,13] width:125 height:27  
  
    on btn8 pressed do  
        (  
            messagebox str4  
        )  
    )  
)  
  
createdialog unnamedRollout
```

最后，就要过年了，愿大家过一个愉快的春节，明年来咱们接着搞脚本，要记得回来啊

本节原帖地址：<http://www.cgplusplus.com/bbs/viewthread.php?tid=326>

本节结束。